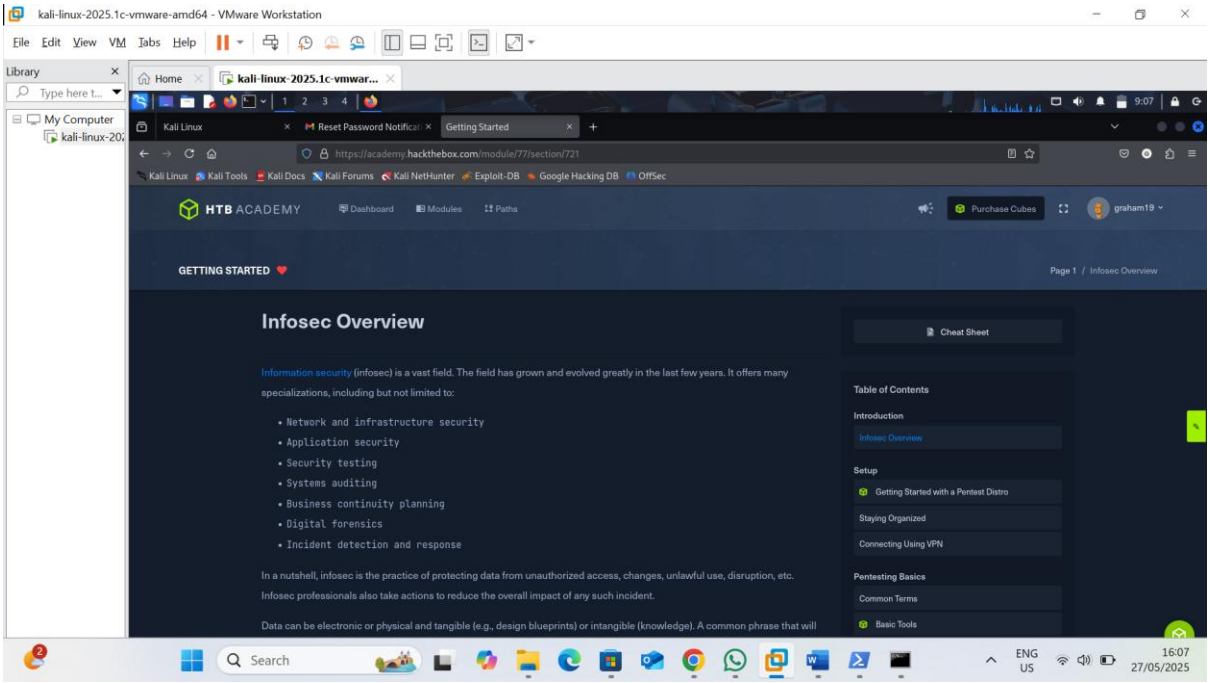
NAME:	**GRAHAM DESCENT OYIGO**
GMAIL:	**Grahamoyigo19@gmail.com**
ADMISSION NUMBER:	**cs-sa10-25023**
MODULE:	**GETTING STARTED ON HACKTHEBOX**
COMPLETION LINK:	https://academy.hackthebox.com/achievement/1921674/77

GETTING STARTED ON HACKTHEBOX

1. INTRODUCTION

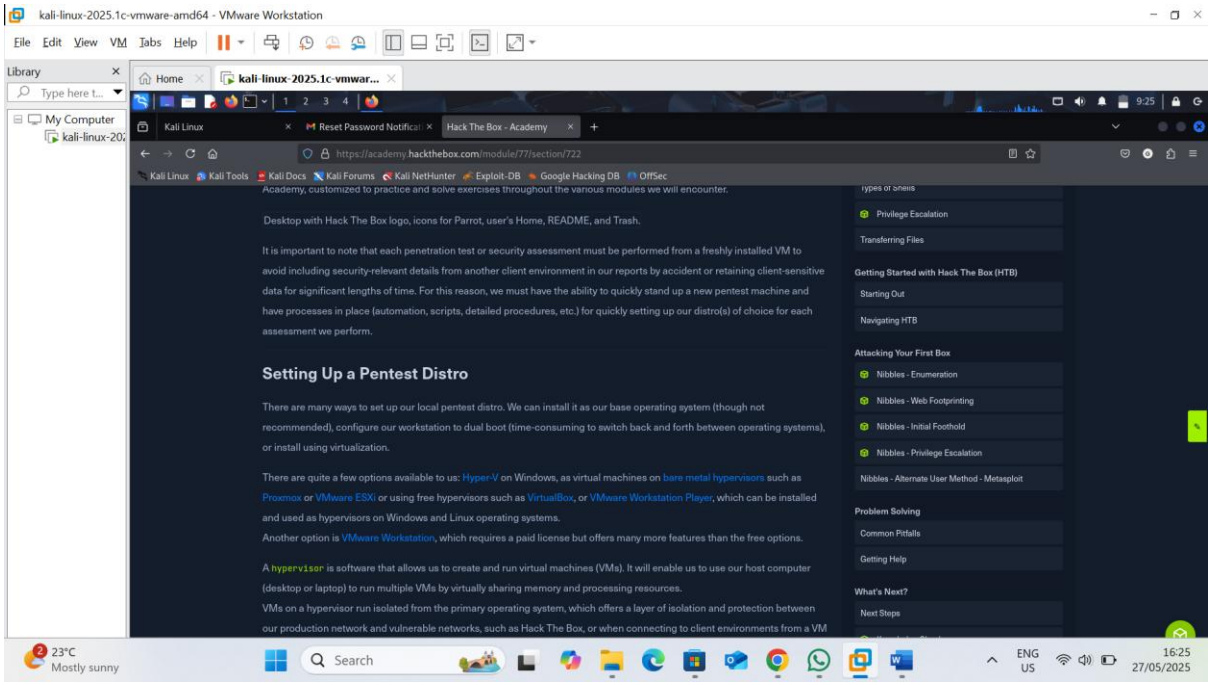
I began this module by learning about information security (infosec), understanding it as a broad field focused on protecting data and emphasizing the "confidentiality, integrity, and availability" (CIA) triad. I then explored the five-step risk management process, which involves identifying, analyzing, evaluating, dealing with, and monitoring risks. I also distinguished between red teams, who act as attackers to find vulnerabilities, and blue teams, who defend against threats, and learned how penetration testers play a crucial role in identifying and mitigating an organization's security weaknesses. This module has provided me with a hands-on introduction to fundamental infosec concepts and penetration testing techniques.



2. SETUP

2.1. Getting Started with a Pentest Distro

Here, learned that penetration testers need comfort with various OS and technologies, especially setting up Linux and Windows attack machines. Choosing a pentest distro is personal, as no single one has every tool, leading some to customize their own. Critical for assessments is starting with a fresh VM to avoid data leakage. Virtualization via hypervisors is preferred for isolation. A home lab is vital for practice. For VM setup, ISOs offer customization, while OVAs allow rapid deployment.

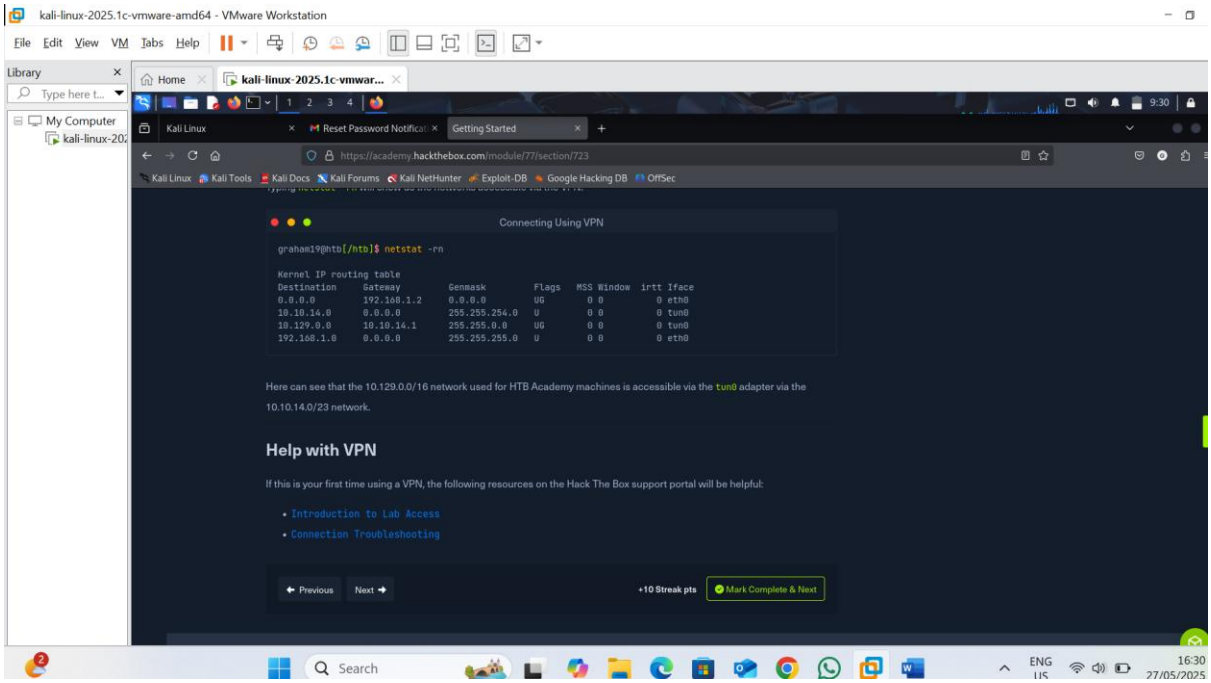


2.2. Staying Organized

In this section, I learned that staying organized with clear, accurate documentation is crucial for all infosec activities, from client assessments to CTFs. I learned I need to maintain a structured folder system on my attack machine to save all assessment data, including scoping, enumeration, exploitation evidence, and credentials. I learned that experimenting with different folder structures is key to finding my most efficient workflow. I learned about various note-taking tools and the importance of choosing one that fits my needs, while ensuring client data remains local. Finally, I learned that building a personal knowledge base with quick reference guides and a vulnerability database will significantly improve my overall efficiency and reporting.

2.3. Connecting using VPN

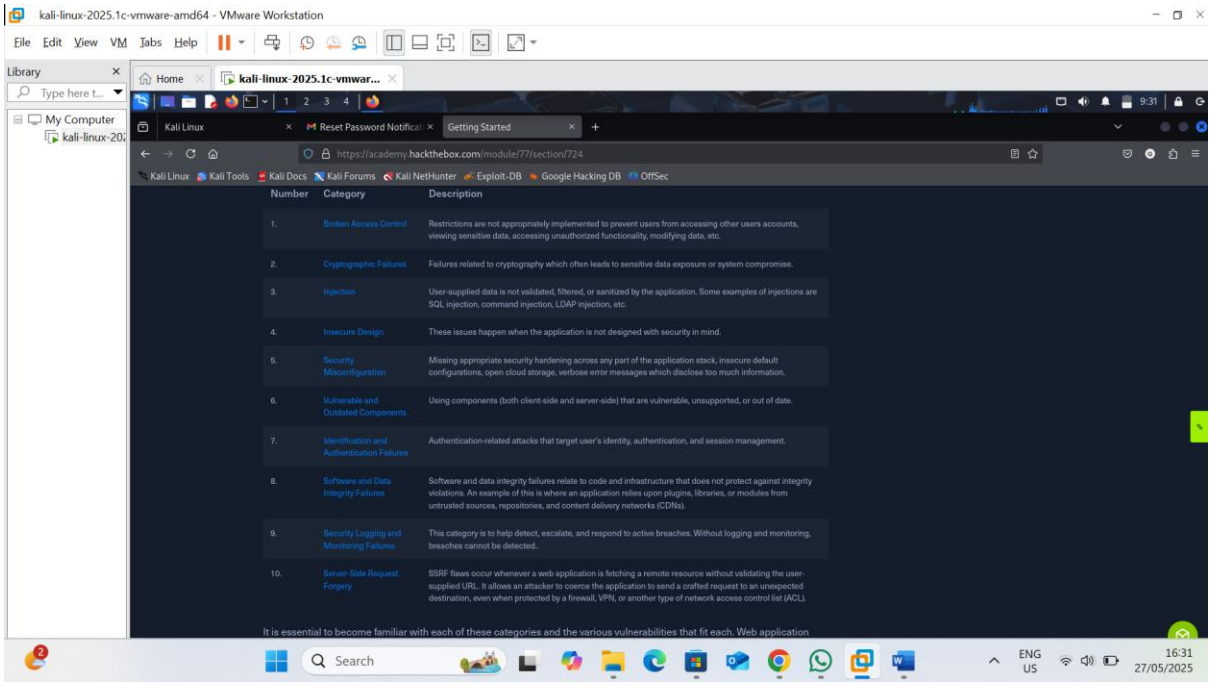
Here, I learned that VPNs provide a secure connection to private networks by encrypting traffic and masking my IP address. I learned that I can connect to HTB labs via VPN, always using a secured VM. The openvpn command, verified by checking for a tun adapter with ifconfig, establishes this connection.



3. PENTESTING BASICS

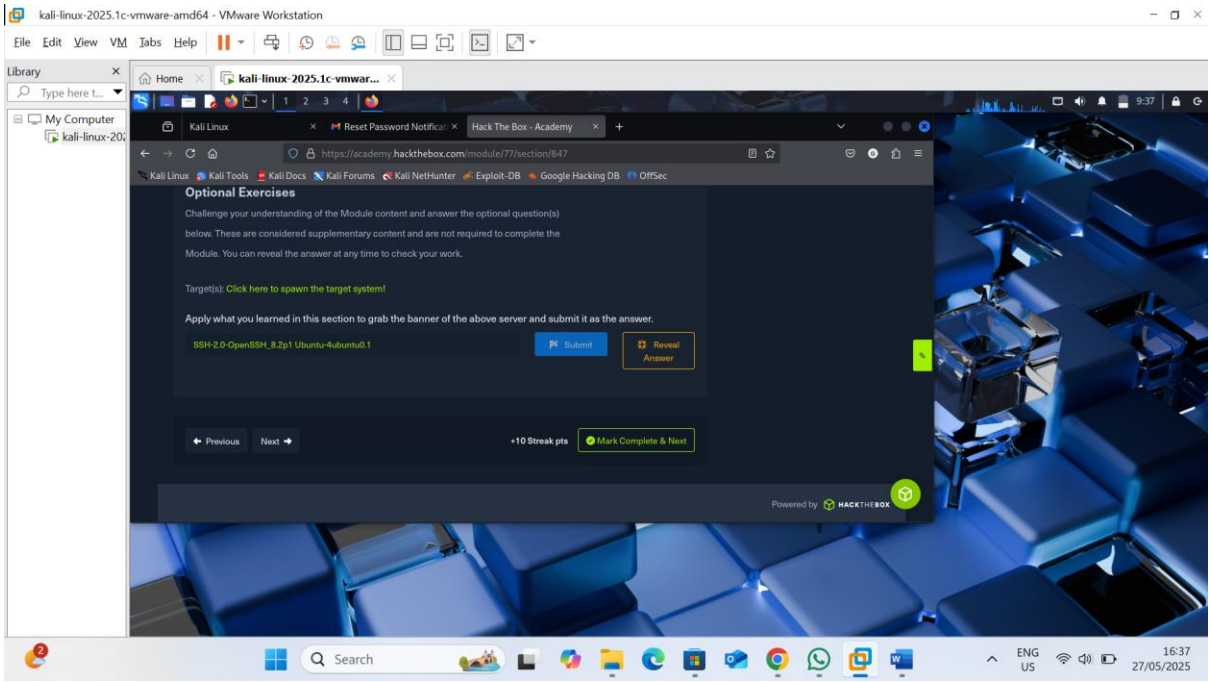
3.1. Common Terms

I learned about common terms in penetration testing, starting with shell, which is a program for interacting with an operating system via keyboard input, like Bash on Linux. I learned that getting a shell means gaining interactive command-line access to a compromised system. I learned about three main shell types: reverse shells, which connect back to my attack machine; bind shells, which listen on the target for my connection; and web shells, which run commands via a web browser. I also learned that ports are virtual network connection points managed by the operating system, associated with specific services (e.g., port 80 for HTTP). I learned about the two categories of ports, TCP (connection-oriented) and UDP (connectionless), and the importance of memorizing common port numbers for efficient enumeration. Finally, I learned that a web server handles HTTP traffic on the back-end and often runs on ports 80 or 443, representing a significant attack surface if vulnerable. I also learned about the OWASP Top 10, a crucial list of the most dangerous web application vulnerabilities.

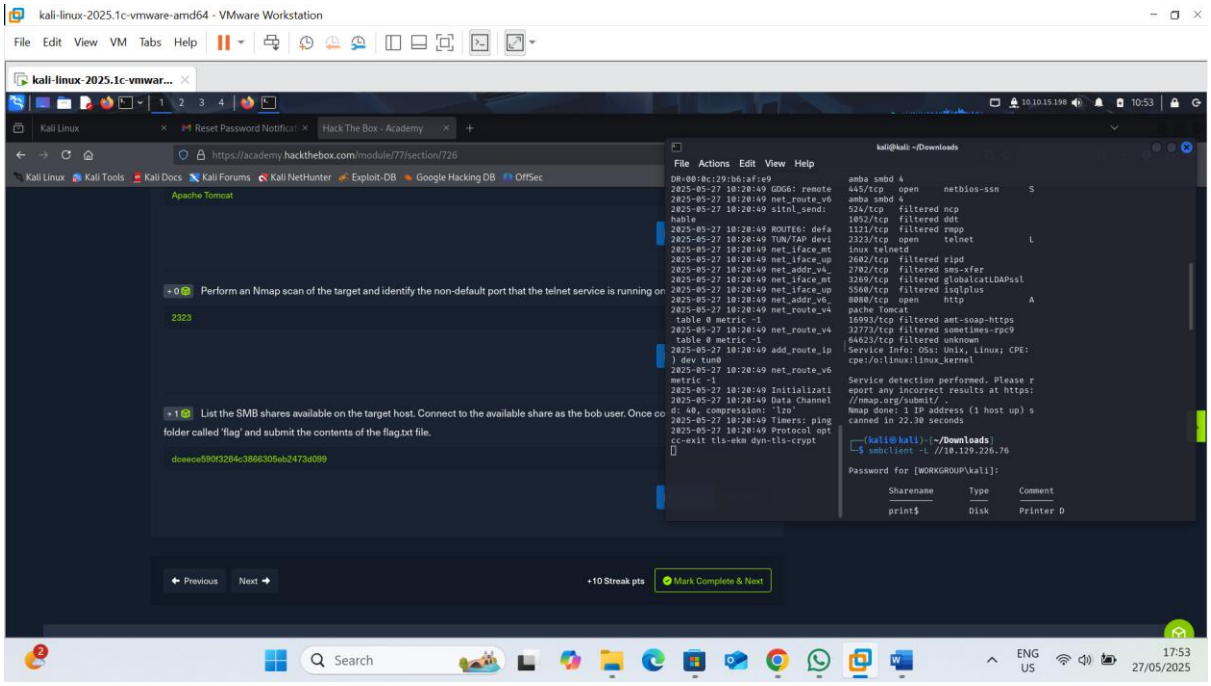


3.2. Basic Tools

I learned about crucial basic tools: SSH for secure remote access and stable connections; Netcat (and Socat) for versatile network interaction, including connecting to shells and banner grabbing; Tmux for efficient terminal multiplexing with multiple windows and panes; and Vim, a powerful, keyboard-driven text editor for remote file editing. Mastering these non-pentesting tools is essential for effective penetration testing.

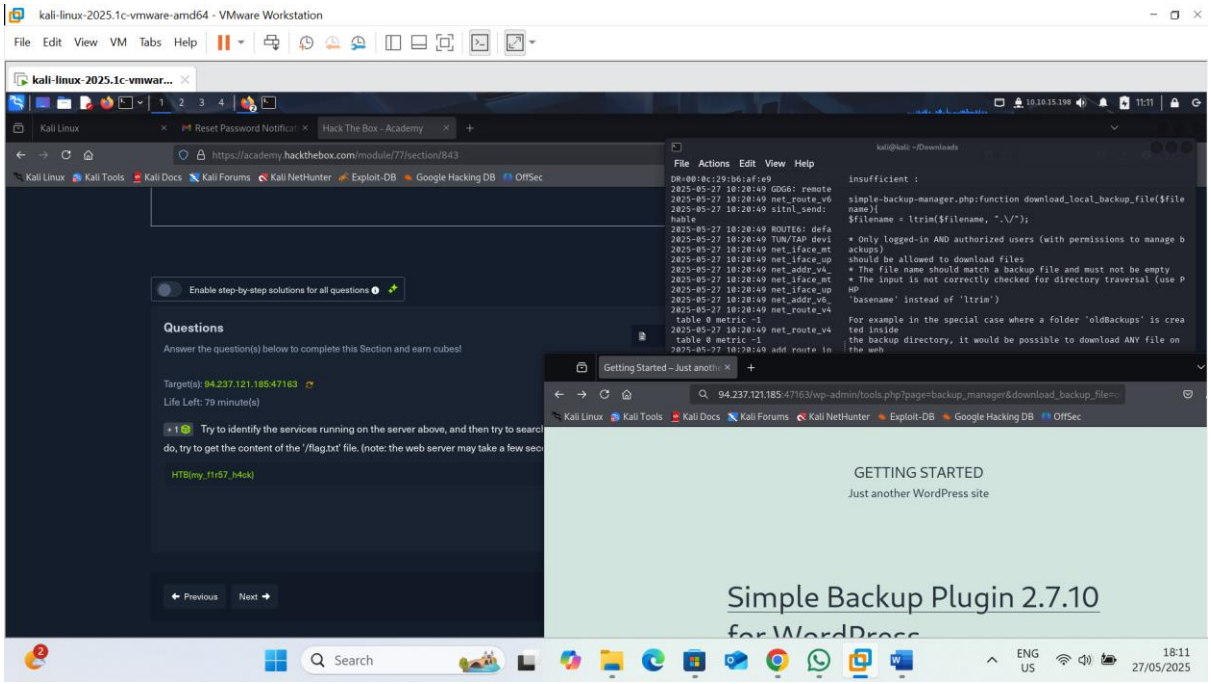


3.3. Service scanning



I learned that initial machine exploration involves identifying the OS and services, primarily using Nmap to scan for open ports and identify versions. I learned to use Nmap for both basic and comprehensive scans, including banner grabbing. I also learned about enumerating specific services like FTP for sensitive data, and SMB for OS details and shares. Lastly, I learned that SNMP can expose device information via community strings

3.4. Public Exploits

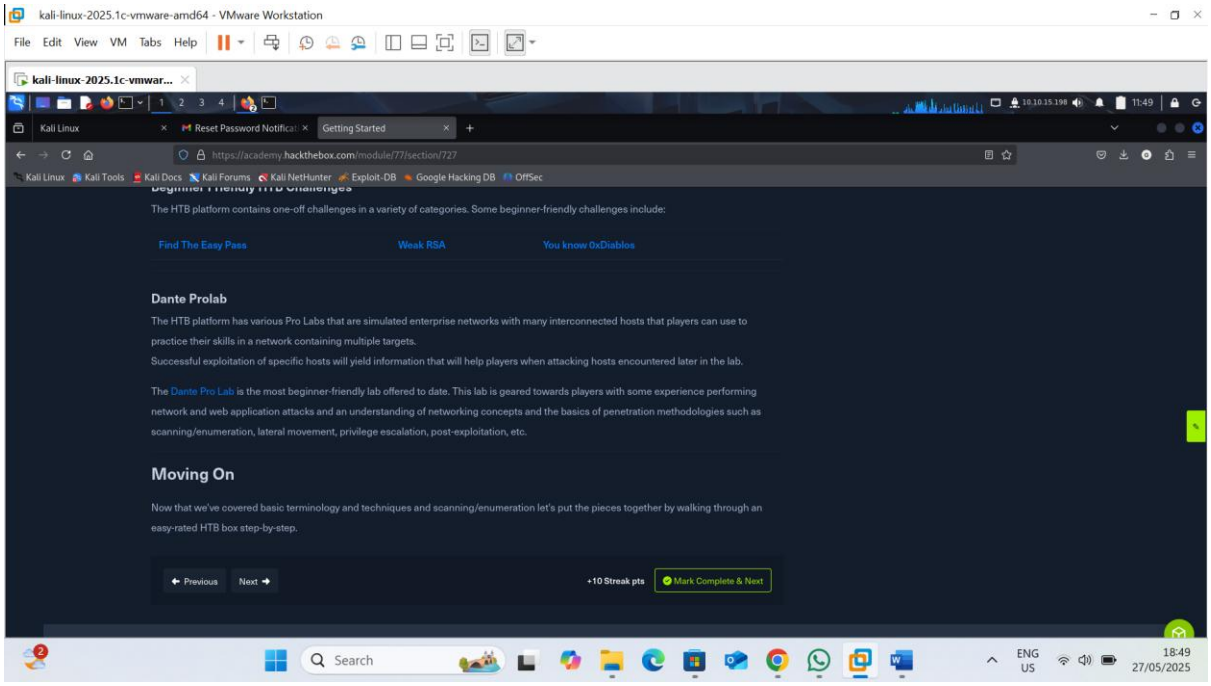


I learned that after identifying services, the next step is to find and utilize public exploits. I learned to use searchsploit to search for known vulnerabilities and online exploit databases

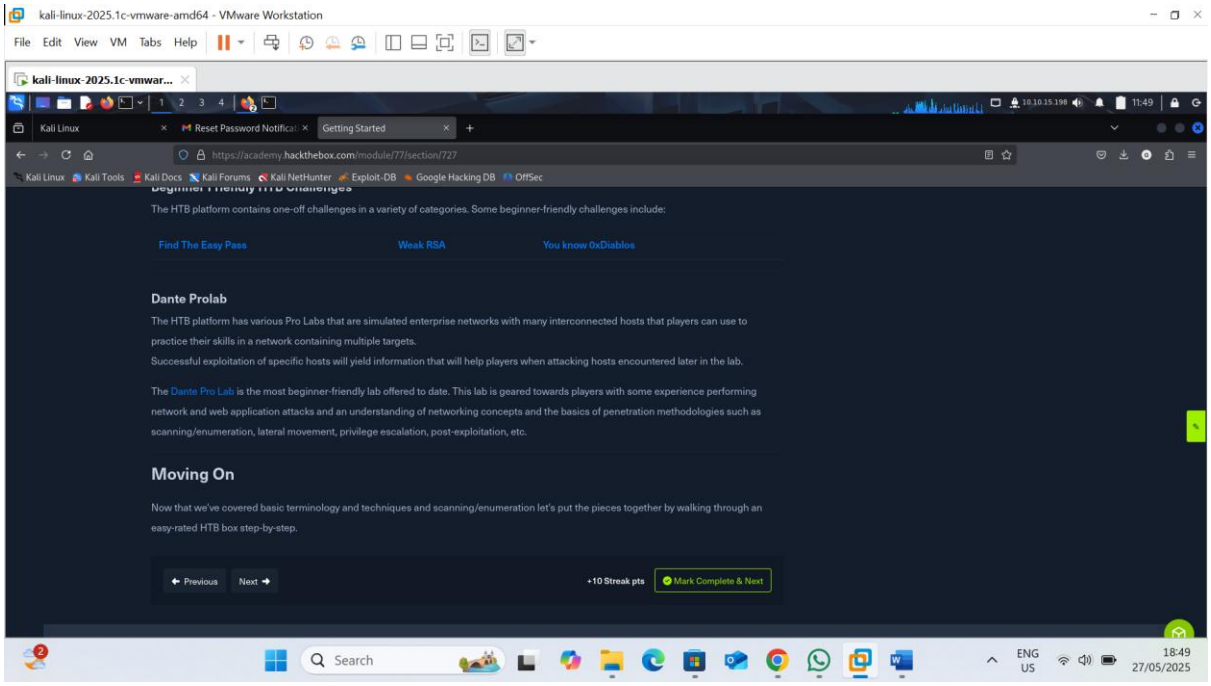
like Exploit DB. I then learned about the Metasploit Framework (MSF), a powerful tool with built-in exploits, reconnaissance scripts, and post-exploitation features like Meterpreter. I learned how to use msfconsole to search for, select (use), configure (set RHOSTS, set LHOST), check vulnerability (check), and execute (exploit or run) exploits, leading to gaining a shell on a compromised system. I also learned that while Metasploit is valuable, I shouldn't rely solely on it and must understand manual techniques too.

4. GETTING STARTED WITH HACKTHEBOX

4.1. Starting Out



4.2. Navigating HTB



5. ATTACKING MY FIRST MACHINE
   5.1. Nibbles – Enumeration

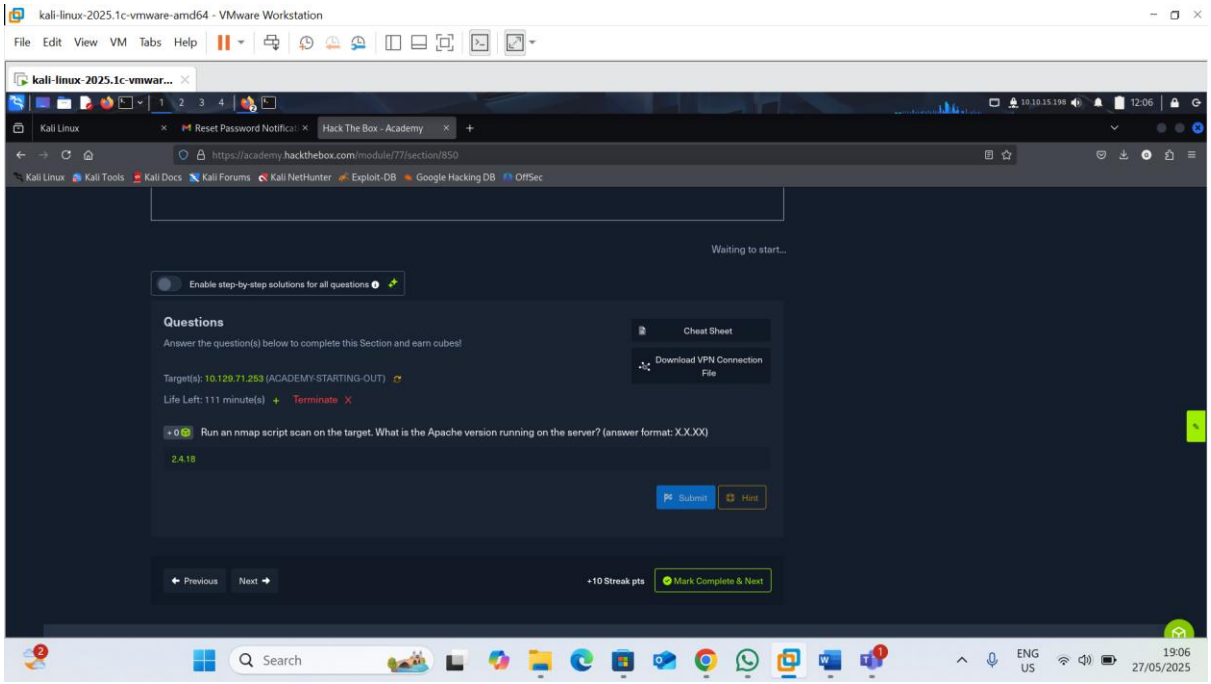
I learned about approaching Hack The Box (HTB) machines, specifically using the "Nibbles" box as an example, which is an easy-rated Linux target. I learned that my initial step is always basic enumeration, and understanding the target's pre-known information (like IP, OS, and attack vector) defines it as a "grey-box" approach, unlike "black-box" scenarios where only an IP is given.

I then learned about the three main penetration testing approaches:

- **Black-Box**: I have little to no prior knowledge of the target, simulating a real attack but potentially missing vulnerabilities.
- **Grey-Box**: I'm given some information, such as IP ranges or low-level credentials, allowing more focus on misconfigurations and exploitation.
- **White-Box**: I have complete access, including source code or administrator credentials, enabling a highly comprehensive analysis for hard-to-discover flaws.

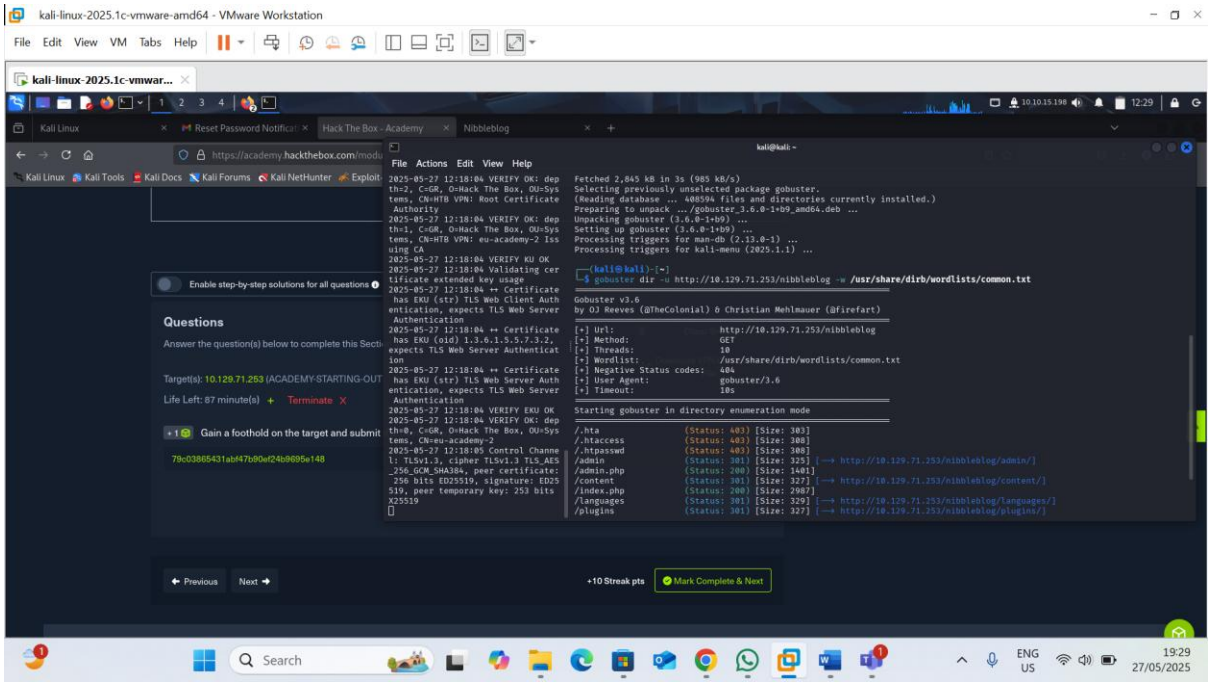
For enumeration, I learned to start with Nmap, using nmap -sV --open -oA nibbles_initial_scan <ip address> to scan for open ports and services, saving the output in multiple formats. I also learned the importance of extensive note-taking and saving all console output to aid in reporting and incident response. I learned that the initial scan on Nibbles revealed an Ubuntu Linux host with an Apache web server on port 80 and an OpenSSH server on port 22. I further learned to perform a full TCP port scan with nmap -p- --open -oA nibbles_full_tcp_scan <ip address> to find any services on non-standard ports, and to use nc -nv for banner grabbing. Finally, I learned to use nmap -sC -p 22,80 -oA

nibbles_script_scan <ip address> for default script scans and nmap -sV --script=http-enum -oA nibbles_nmap_http_enum <ip address> to enumerate common web directories, although these specific scans didn't yield immediate results on Nibbles.

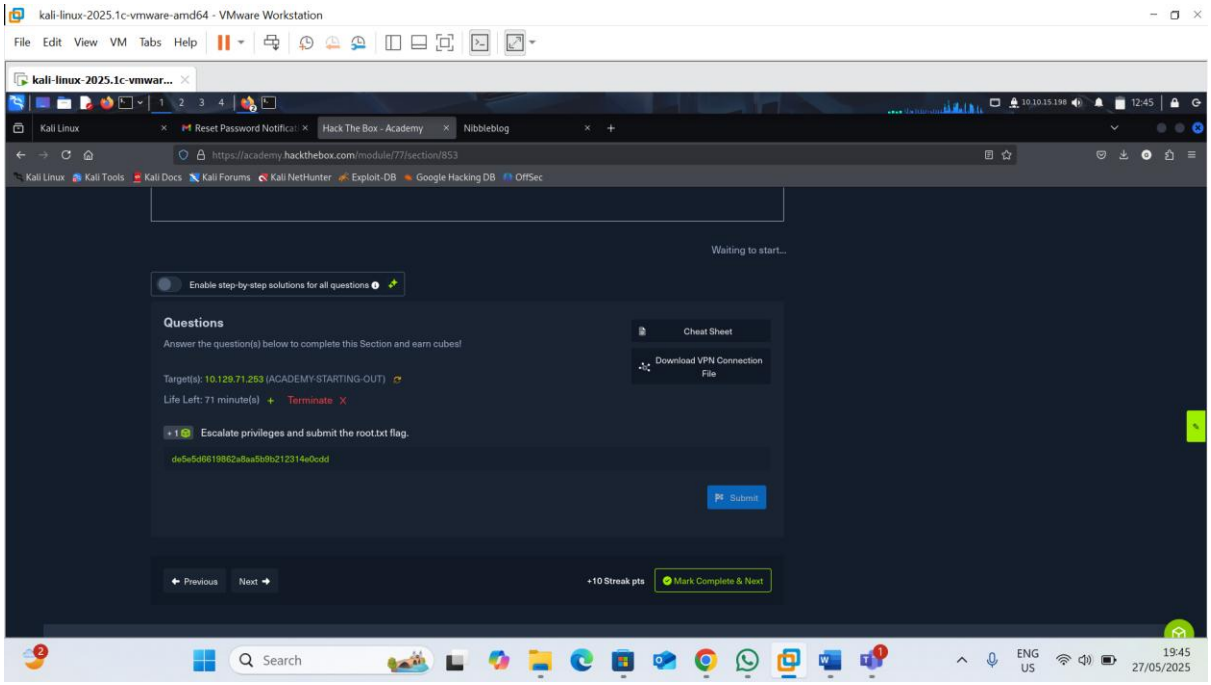


5.2. Nibbles - Initial Foothold

I learned that exploiting the Nibbles machine involves gaining an initial foothold through its web application's admin portal. This is achieved by uploading a PHP code snippet via the "My image" plugin, confirming code execution by accessing the uploaded file, and then replacing the snippet with a reverse shell to gain access as the nibbler user. Finally, the shell is upgraded for better functionality, and the user flag is retrieved.



5.3. Nibbles - Privilege Escalation



I learned that privilege escalation on Nibbles involves leveraging a world-writable script that the nibbler user can execute with sudo privileges without a password.

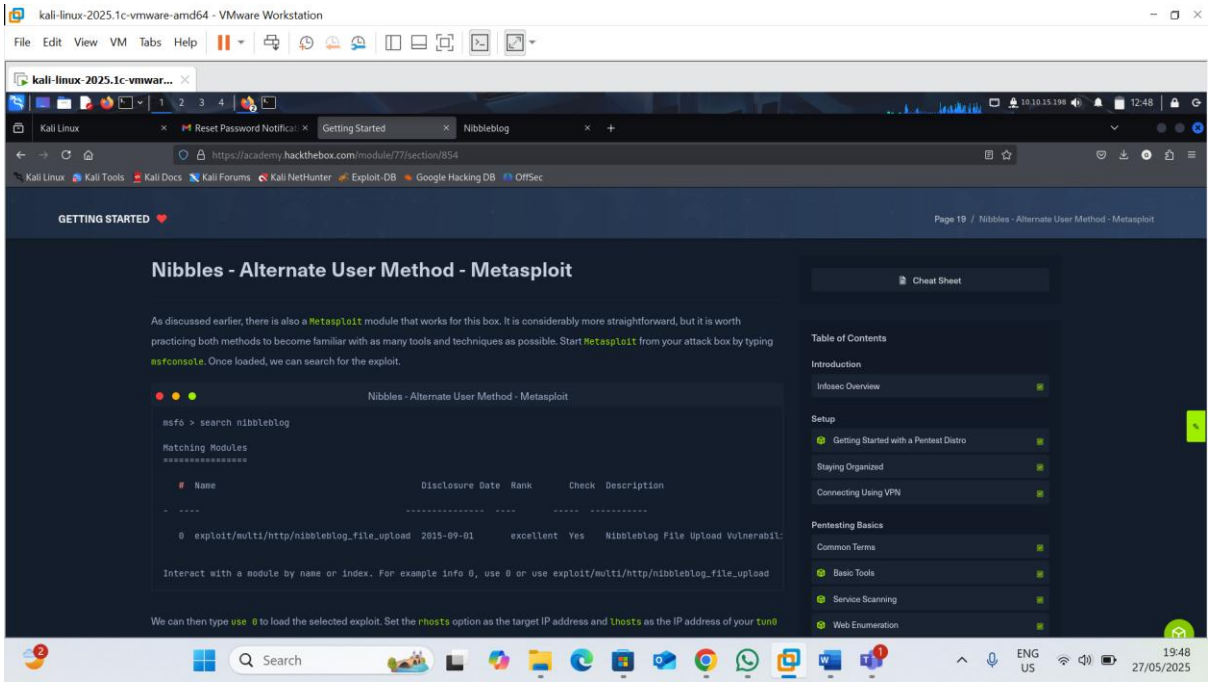
The process involves:

1. **Unzipping personal.zip**: This reveals monitor.sh, a shell script owned and writable by the nibbler user.

2. **Automated Enumeration with LinEnum.sh**: I learned to transfer and run LinEnum.sh to quickly identify privilege escalation vectors. This script specifically highlighted that nibbler can run /home/nibbler/personal/stuff/monitor.sh as root using sudo without needing a password.
3. **Exploiting the Writable Script**: By appending a Bash reverse shell one-liner to the monitor.sh script, I can execute it.
4. **Gaining Root Shell**: When sudo /home/nibbler/personal/stuff/monitor.sh is executed, the appended reverse shell connects back to my Netcat listener, granting me a root shell (uid=0(root)).
5. **Retrieving the Root Flag**: From the root shell, I can then retrieve the root.txt flag.

I also learned the importance of note-taking and using various tools to achieve the same objectives.

5.4. Nibbles - Alternate User Method – Metasploit



I learned about an alternate method to gain a user shell on the Nibbles machine using Metasploit, which offers a more streamlined approach compared to manual exploitation. I learned to start msfconsole and search for "nibbleblog" to find the relevant exploit module, exploit/multi/http/nibbleblog_file_upload. I then learned to load this module, set the RHOSTS (target IP) and LHOST (my attacking IP), and provide the USERNAME (admin) and PASSWORD (nibbles) for authentication, along with setting TARGETURI to nibbleblog. I also learned to change the payload to generic/shell_reverse_tcp before executing the exploit, which resulted in a command shell session as the nibbler user. This confirms that multiple tools and techniques can achieve the same result, and it's valuable to practice various

methods. From this point, I learned that the privilege escalation path remains the same as the manual method.

6. CONCLUSION

This module has provided me with a foundational understanding of penetration testing, from initial setup to gaining a foothold and escalating privileges on a target system. I've learned that thorough enumeration is an iterative and critical first step in any assessment, whether it's a black-box, grey-box, or white-box engagement. I've grasped the importance of using tools like Nmap for port and service scanning, followed by detailed web footprinting with tools such as WhatWeb and Gobuster to uncover hidden files and directories.

I've also learned the value of leveraging public exploits found via searchsploit or online databases once vulnerabilities are identified. A key takeaway is the process of gaining an initial foothold, including techniques like exploiting file upload vulnerabilities to achieve remote code execution and ultimately obtaining a reverse shell. Furthermore, I've learned the practical step of upgrading a basic shell to a fully interactive TTY for better functionality, often using a Python trick. Finally, I've understood that privilege escalation involves meticulous manual and automated enumeration of the file system using scripts like LinEnum and LinPEAS, looking for misconfigurations, vulnerable services, and sensitive data to achieve root access. The experience with the Nibbles box reinforced that multiple methods often exist for both gaining a foothold and escalating privileges, emphasizing the need to explore and understand various tools and techniques while maintaining detailed notes throughout the process.